CarFile

Final Report

STRiNg, Int. Team Members

- *Steven Bock* – sbock@mail.smu.edu
- *Travis Siems* – tsiems@mail.smu.edu
- *Ryan Sligh* – rsligh@mail.smu.edu
- *Nicholas Roberts* – robertsn@mail.smu.edu

Table of Contents

Executive Summary

This report includes all of the information related to the final handover of the CarFile project. We have successfully created a release version of the CarFile Automobile Maintenance Logging Application. The goal of the CarFile project is to create a cross platform Android and iOS application that enables users to easily keep track of maintenance tasks across all of their automobiles.

Understanding the importance of user feedback in the development process, we adopted certain processes to facilitate early testing. In order to allow for user feedback throughout development, we adopted an iterative model of development. This allowed us to continuously test our application, look for current and potential problems, gather feedback, and use our results to influence our next steps in development.

We have brought CarFile to a level of functionality and stability suitable for release on both the iOS and Android platforms. The series of prototypes we created in each iteration allow extensive usability testing and contributed to the successful completion of the project. The user interfaces on Android and iOS have almost the same functionality, however the implementation looks somewhat different in order to meet the design standards users have come to expect on both platforms.

As we continue on to the maintenance phase, we can release updates swiftly and safely because of the solid foundation we have already created for our app. The user interface on both platforms may diverge as design standards change with new OS versions. We are proud to present CarFile to our client. This report will touch on each element of development of this project over the semester, including future development.

Introduction to the CarFile Project

Introduction

The purpose of the CarFile project is to design and implement a cross platform app that allows individuals to quickly and easily log and view information about their automobile maintenance history.

Report Summary

This report will serve 3 purposes:

i. Give stakeholders an update on the progress made during development.
ii. Increase the understanding of the requirements that we have decided the project needs to meet.
iii. Increase understanding of certain decisions regarding design and implementation and why they were made.

Project Details Description

The goal of the CarFile project is to produce a cross-platform iOS and Android application that enables users to intuitively log maintenance information using a well designed interface that assists the user in the process of tracking maintenance records and important dates regarding their automobile(s).

Application Functions

Upon first starting CarFile, users will be able to set up a vehicle profile that may include any essential basic information about their car. After the user inputs basic information about their car, they will have a chance to either select that car or add another vehicle to the app. Once the user has completed the input of basic vehicle information, he or she can then select one of the car profiles to lead them to another page containing a list of basic car maintenance types. From this page, the user can select one of the maintenance types to view previously inputted maintenance records and dates. The user can also choose to input a new piece of maintenance information at this time. If the user elects to add new information, the user will be presented with a template to help ease the process of adding maintenance information.

Functional Requirements

The requirements have not changed very much since the last iteration. Data generated by the user will be stored in a relational database. We will use a common data structure between the iOS and Android versions of CarFile so data exported from either of the version can be imported to the other version with ease.

When operating the app, the user must be able to:

- Add car objects with basic information on the car
- Add preventative maintenance information for a car
- Edit the basic information of a car
- Edit preventative maintenance information
- View cars and preventative maintenance information
- View alerts and notifications for dated preventative maintenance information

Non-Functional Requirements

The below requirements are not requirements of functionality, but of the quality of that functionality.

The CarFile team remains committed to respecting our users right to privacy for their data. We do not collect, sell, or analyze any user data. In fact, there is currently no planned component that interacts with the network, giving us no way to view CarFile users' data, even if we wanted to. This lack of network interaction also means that the app will be available at all times of the day and anywhere in the world because it is not dependent on servers being kept operational. CarFile is also designed to be lightweight and fast, performing well on a wide array of devices.

Organizational Requirements

The CarFile team must have a clear line of communication through our GroupMe channel. Steven, the project manager for Milestone I, led the preliminary requirements gathering and provisional design phase. He kept our scope, skillsets, and resources in mind when leading this section of development. Travis, the project manager for Milestone II, delegated tasks effectively and timely, making sure that our goals and deadlines were completed on time. He reviewed the team’s progress before we moved on to the next phase so the team could continue with refreshed organization. Due to current skillsets and technology resources, Travis and Steven headed the iOS development, and Nick and Ryan headed the Android development. After we did some requirements gathering and provisional design work, we implemented our design into a working prototype on the respectively assigned devices.

Nick, the project manager for Milestone III, led our group effectively and helped keep us on time in terms of requirements gathering, design, prototype development and usability testing. Tasks were delegated in a way that helped our team complete its goals in a timely manner. Leading into the final iteration, we are in a good place to complete the project’s requirements. Working through challenges and learning new information has helped develop the project and our team throughout all the iterations so far.

User Interface and Usability Requirements

Since CarFile may potentially require users to enter a considerable amount a data if they own many vehicles, wherever possible, we aim to minimize text entry. This includes using Edmond's API to autofill car information based on it's VIN number or using a date slider for inputting temporal data. Through user testing, we found which areas are time consuming to fill out and then apply solutions to those areas.

Usability Testing

When thinking about how we would hold testing sessions, we decided to consider two groups of potential users: avid car users and casual car users. We define an avid car user as someone who does some of their own maintenance for their car(s), and we define a casual car user as someone who pays mechanics to do preventative maintenance on their car(s). Initially, we did white box testing ourselves to find simple bugs and issues, and then did black box testing to get insight as to how intuitive our interface and application as a whole actually is. For the black box testing, we held some self-driven testing where we observed users interact with our application without interfering. After users figured out the application, we gave them some basic tasks, such as “Log a maintenance item using a receipt for some maintenance you just had for your car” and “Edit your basic car information to update you license plate” and other scenarios that would use all the functionality of the application. These scenarios helped our users understand when to use our application and see how it could be useful.

The usability testing sessions for the CarFile gave the team insight into how to improve the app. Most of the testing was performed on the Android version of CarFile. Most testers did not think to slide to the right to display the slide out car menu. This menu is an important part of app navigation and should be easier to find. We removed the gesture based controls from the interface design. The car menu is now displayed first when the app opens, and it can be access to by pressing the back button on the top of the screen to move up the page hierarchy until you reach the top menu. The main source of frustration was data input, namely VIN number input. We use Edmund's API to return car information based off the VIN number inputted, but if the user mistypes their VIN, we neglected to display a "Car not found" message when Edmund's API fails to return results. Also, if the user did type their VIN number correctly, there sometimes a three to eight second pause as the app waits for a response from Edmund's API. Users noticed this lack of responsiveness and many thought the application may have crashed during this delay. Some users pressed the continue button multiple times, which revealed a bug where two or more pages will load once Edmund's API returns the data. We will need to display a visual indicator to indicate the app is processing and has not crashed. Also, users did not like that all fields in the car information edit page were required fields. Most of these problems can be easily fixed. Users were able to edit car information and delete cars without issue, but would prefer a warning message before deleting any data. Many of these suggestions have helped define what we need to do for our final iteration.

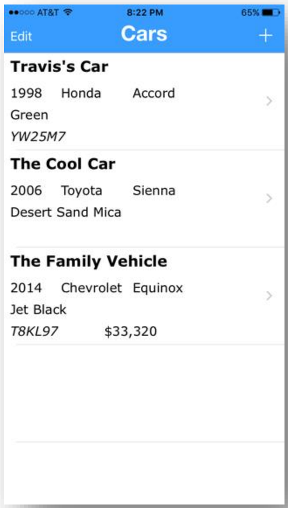
Final Iteration and Future Development

In the final iteration, we finished implementing the required functionality in both the iOS and Android versions, performed additional usability testing, and made more enhancements to the user interface. Below, we sample pages from both the iOS and Android interfaces to give the reader a feel for what the application looks like and the cross platform differences and similarities.

Comparing Android and iOS Versions

The iOS and Android versions include minor differences in appearance, as we had to optimize for particular devices. Some slight functional differences allow us to see which functionality our users want for the fully functional application and which our users do not care for. The Android version allows the user to upload a picture to be the profile picture of the car, while the iOS version currently pulls a picture from the Edmund's API using a VIN number. The iOS version includes Alert functionality to help the user see which car and maintenance items need updating. Although there are these minor differences, the main functionality is the same in both, and users will be able to navigate easily in either version.

Final iOS Interface

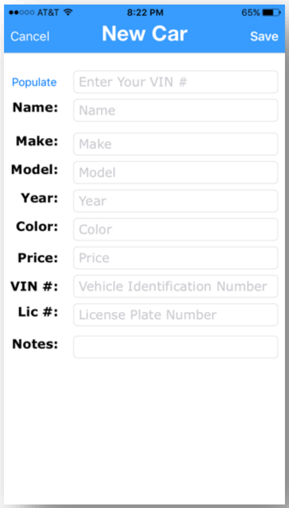

●●○○○ AT&T 8:22 PM 65%

Edit **Cars** +

Travis's Car
1998 Honda Accord
Green
YW25M7

The Cool Car
2006 Toyota Sienna
Desert Sand Mica

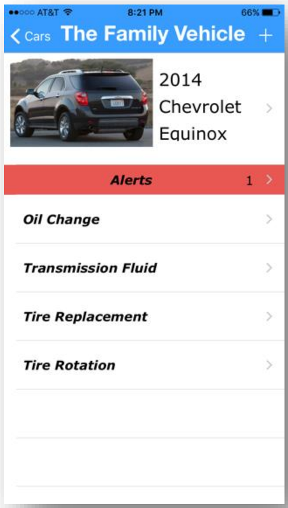
The Family Vehicle
2014 Chevrolet Equinox
Jet Black
T8KL97 $33,320


●●○○○ AT&T 8:22 PM 65%

Cancel **New Car** Save

Populate	Enter Your VIN #
Name:	Name
Make:	Make
Model:	Model
Year:	Year
Color:	Color
Price:	Price
VIN #:	Vehicle Identification Number
Lic #:	License Plate Number
Notes:	

Above from left to right: *iOS* Home Screen, New Car Page.

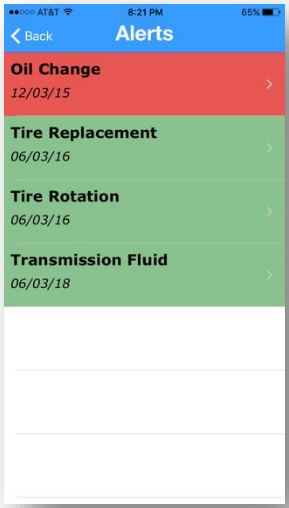

●●○○○ AT&T 8:21 PM 66%

Cars **The Family Vehicle** +

2014
Chevrolet
Equinox

Alerts 1

Oil Change
Transmission Fluid
Tire Replacement
Tire Rotation


●●○○○ AT&T 8:21 PM 65%

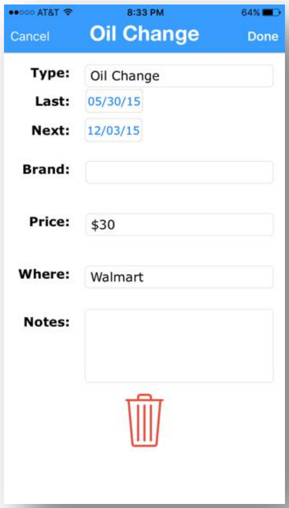
Back **Alerts**

Oil Change
12/03/15

Tire Replacement
06/03/16

Tire Rotation
06/03/16

Transmission Fluid
06/03/18

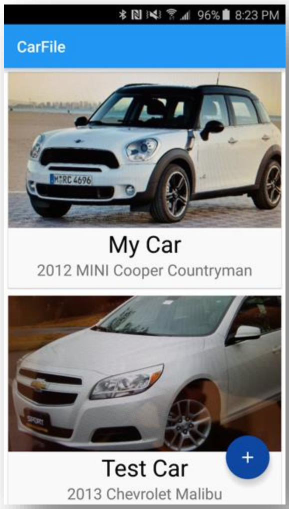

●●○○○ AT&T 8:33 PM 64%

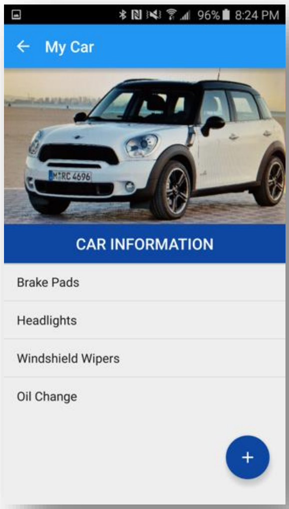
Cancel **Oil Change** Done

Type:	Oil Change
Last:	05/30/15
Next:	12/03/15
Brand:	
Price:	$30
Where:	Walmart
Notes:	

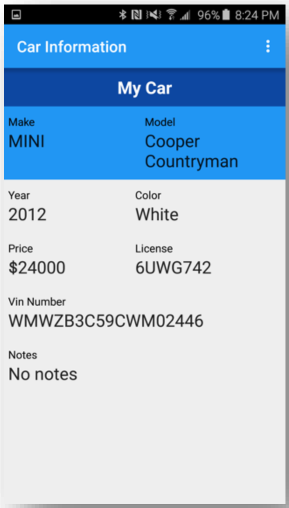
Above from left to right: *iOS* Car Page, Alerts Page, Edit Maintenance Information

Final Android Interface





Above from left to right: *Android* **Home Screen, Car Page**

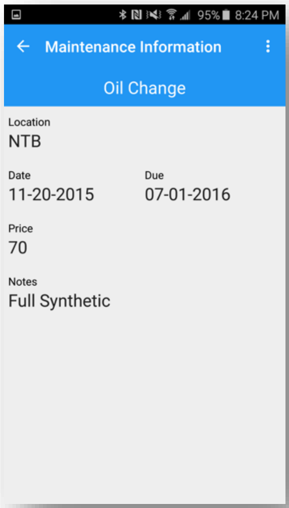


My Car

Make MINI	Model Cooper Countryman
Year 2012	Color White
Price $24000	License 6UWG742

Vin Number
WMWZB3C59CWM02446

Notes
No notes



Oil Change

Location
NTB

Date 11-20-2015	Due 07-01-2016

Price
70

Notes
Full Synthetic

Above from left to right: *Android* **Car Information, Maintenance Information**

Future Development

While the requirements set out at the beginning were met, some features that our team was planning on had to be cut in order to deliver the project within the span of this class. In the future, some of this functionality will be added. These include customizable reminder intervals, maintenance information history, a more adaptable UI suited for extra user input, and additional platforms such as Android Wear and Windows 10. Also, while a key requirement of our application is that it be functional without an internet connection, we realized that having a cloud based back-end would allow us to deliver more advanced functionality. In the future, this can be used to allow for advanced features like synchronized logs, extra information from the VIN number, a web based interface, and transfer of maintenance history to new car owners. We realize that a weakness of our program is the great amount of manual entry, so future development will include more ways to optimize the user interface, automatically pull information using APIs, and potentially including receipt reader functionality. When we add all these extra features, we'll want to monetize our app to make that time spent worthwhile. We will analyze the market to help decide if we should monetize CarFile with advertisements or a purchasable "pro" version of the app with additional features.